REMARKS/ARGUMENTS

This Amendment and the following remarks are intended to fully respond to the non-final Office Action dated June 30, 2005 (hereinafter the Action). In the Action, claims 1-46 were examined, and all claims were rejected. In this Response, claim 1 has been amended. Reconsideration of these rejections, as they might apply to the original and amended claims in view of these remarks, is respectfully requested.

Statement of Common Ownership under 35 U.S.C. § 103(c)

The present patent application (Serial No. 09/613,032) and U.S. Patent No. 6,345,361 to Jerger, et al. (hereinafter *Jerger*) were, at the time the claimed invention in the present patent application was made, owned by, or subject to an obligation of assignment to, Microsoft Corporation of Redmond, Washington.

Claim Rejections – 35 U.S. § 103

The Action rejected claims 1-12, 20-25, and 28-36 under 35 U.S.C. § 103(a) as being unpatentable over U.S. Patent No. 6,138,238 to Scheifler, *et al.* (hereinafter *Scheifler*) in view of U.S. Patent No. 6,526,513 to Shrader, *et al.* (hereinafter *Shrader*) and further in view of U.S. Patent No. 6,735,758 to Berry, *et al.* (hereinafter *Berry*). Applicants respectfully traverse these rejections, in part because the proposed combination of references does not teach or suggest every claim element. In addition, there is no valid motivation to combine the references. Each of these issues is covered below.

Independent claim 1 has been amended in order to clarify the subject matter. No new subject matter has been added. Amended claim 1 provides, *inter alia*, a method of determining whether a requested permission is satisfied within a runtime call stack comprising:

> associating a first permission grant object with a first code assembly in the runtime call stack;
> dynamically overriding the set of permissions that is assigned to a second permission grant object associated with a second code assembly preceding the first code assembly;
> creating a permission request object within the called code frame to demand the requested permission;

demanding via the permission request object the requested permission from the first permission grant object to allow the called code frame to perform the protected operation;

determining whether the requested permission is provided in association with the first code assembly by the first permission grant object, responsive to the demanding operation; and

permitting the execution of the called code frame to perform the protected operation, if the requested permission is provided in association with the first code assembly, whereby a full walk of the runtime call stack may be avoided.

In rejecting the cited claims, including claim 1, the Action stated:

Scheifler discloses an access control apparatus utilizing a security policy file, a call code stack, and an executor ... Scheifler does not disclose the dynamic request/grant of a permission set. However, Shrader and Berry discloses a method, computer program product of determining whether a requested permission, wherein the permission is at least one of a set of permissions, requested by a called code frame, is satisfied within a runtime call stack so as to allow the called code frame to perform a protected operation, the method comprising:

b) dynamically overriding the set of permissions that is assigned to a permission grant object associated with at least one other code assembly preceding the first code assembly; (see Shrader col. 2, lines 12-17; col. 3, lines 49-53; col. 5, lines 29-37: dynamic permissions set, no more checks (i.e., stack walking) made, permission checks terminated; see Berry col. 20, line 64 - col. 21, line 4: parameters for termination of a stack walk, stack walk terminated.)

Applicants respectfully traverse the arguments of the Action. The combination of *Scheifler*, *Shrader*, and *Berry* does not teach or suggest every element of the independent claims, and the claims which depend therefrom. The Action notes that Scheifler does not disclose the dynamic request and grant of a permission set. To fill this hole, the Action cites *Shrader*. *Shrader* relates to an architecture for extending the Java security model to allow a user or administrator to grant permissions on the fly. *Shrader* allows Java applets and applications to prompt a user to grant a permission that does not exist in the Java policy file. In *Shrader*, overrides are stored in one or more files, and not in the stack itself. As a result, all classes in a stack are checked regardless of whether the user has supplied an overriding value. This can clearly be seen in Fig. 3 of *Shrader*, where regardless of whether a user dynamically grants permission, every class in the stack is still checked.

The Action further cites *Berry* to provide for terminating a stack walk. *Berry* provides multiple processor computer systems a method for symmetric multiprocessing (SMP) profiling using a global value of a metric variable with support across multiple systems. The specific paragraph cited by the Action provides:

> A determination is made as to whether the index is equal to zero (step 1508). If the index is equal to zero, the process then returns to determine whether additional call stack unwind trace records are present for processing (step 1510). If additional call stack unwind trace records are present, the process then returns to step 1500 to read another call stack unwind trace record. Otherwise, the process terminates.

Although the cited portion of *Berry* uses the words "call stack" and "terminate," it does not provide for halting a stack walk as set forth in claim 1. *Berry* states here that the invoked process will continue "[i]f additional call stack unwind trace records are present." This means that a complete stack walk occurs, so long as there are records in the stack. *Berry* does not in fact teach or suggest any of the elements of the independent claims.

In order to provide a motivation to combine the cited references, the Action provides the following reasoning:

> *It would have been obvious to one of ordinary skill in the art at the time the invention was made to modify Scheifler to enable the capability of the dynamic permission request/grant as taught by Shrader, and to setup parameters for the termination of a stack walk as taught by Berry. One of ordinary skill in the art would be motivated to modify Scheifler in order to enhance the execution of computer code based on permissions sets (see Shrader col. 2, lines 7-17: "*...architecture for extending Java security model to allow a user ... grant permissions dynamically ... If the user grants the permission, the present invention grants the permission for the ProtectionDomain ...*"), and to employ Berry in order to optimize and accurately maintain profiling information. (see Berry col. 3, lines 36-39: "*... provide a system in which accurate profiling information could be maintained globally for a plurality of processor rather than at the processor level ... *").*

Applicants respectfully submit that the motivations provided are not proper. Although the Action provides portions of the references which provide reasoning for some combination, the reasoning supplied does not justify the combinations supplied here. For example, the motivation supplied for combining *Berry* specifically has no relevance to either *Scheifler* or *Shrader*. Profiling of processor resources is required in multi-processor development

environments, and has no direct relevance to determining access permissions. *Berry* explains this at col. 2, lines 43-46, "This technique is based on the idea of periodically interrupting the application or data processing system execution at regular intervals, so-called sample-based profiling." The reasoning stated by the Action, nor any other portion of *Berry*, offers a valid motivation to combine the reference with *Scheifler* and/or *Shrader*. Moreover, as stated above, the improper combination of *Scheifler*, *Shrader*, and *Berry* does not teach or suggest every element of the independent claims, let alone those of the dependent claims.

With specific regard to the rejected dependent claims, the additional claim elements which they embody are not disclosed in any of the references. For example, claim 6 provides, *inter alia*:

> permitting execution of the called code frame to perform the protected operation, if the requested permission is a subset of the specified permission.

In rejecting this element of the claim, the Action cites the following passages from Shrader (col. 2, lines 12-17; col. 3, lines 49-53):

> *The inventive architecture allows Java applets and applications to dynamically prompt the user to grant a permission that does not exist in the Java policy file. If the user grants the permission, the present invention grants the permission for the ProtectionDomain to which the class asking for the permission belongs.*
>
> *To implement the present invention, a Java Runtime Environment (JRE) is installed that supports dynamic and/or denied permissions via the permissions module. One or more policy configuration files are installed that list the granted or denied permissions.*

Nowhere in the cited portions of *Shrader*, nor anywhere else in the reference, are the elements of claim 6 disclosed. There is no determination of whether "the requested permission is a subset of the specified permission." *Shrader* merely allows a user to override all permissions using a dynamic request. Applicants respectfully request that the rejection of claim 6, and of the other rejected claims, be withdrawn.

The Action rejected claims 13-19, 26, 27, and 37-46 under 35 U.S.C. § 103(a) as being unpatentable over *Scheifler* in view of *Jerger*. In view of the Applicants' above Statement of Common Ownership under 35 U.S.C. § 103(c), *Jerger* may not be used to preclude patentability

under § 103. Applicants therefore respectfully request that the rejections of claims 13-19, 26, 27, and 37-46 be withdrawn.

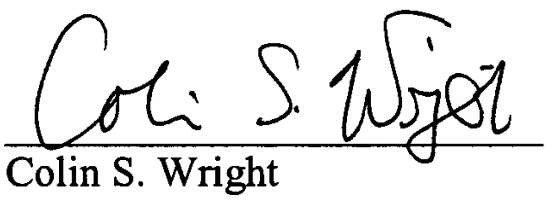
CONCLUSION

In light of the above remarks and amendments, it is believed that the application is now in condition for allowance and such action is respectfully requested. Should any additional issues need to be resolved, the Examiner is requested to telephone Applicants' undersigned representative to attempt to resolve those issues. It is believed that no further fees are due with respect to this filing. However, the Commissioner is hereby authorized to charge any deficiency or credit any overpayment with respect to this patent application to deposit account number 13-2725.

Respectfully submitted,

MERCHANT & GOULD, LLC

Date: September 30, 2005



Colin S. Wright
Reg. No. 57,202

MERCHANT & GOULD, LLC
P.O. Box 2903
Minneapolis, Minnesota 55402-0903
(404) 954-5100

K:\Clients\40\40062\0065-US-04\Resp & Amend to 6.30.05 OA.doc